Amendments to the Claims:

This listing of claims replaces all prior versions and listings of claims in the application:

Listing of Claims:

1. (Currently Amended) An apparatus comprising:

a network interface operative to receive frames, each frame including a header and a payload, the payload having a first bit adjacent the header;

an alignment module operative to append a dummy portion to the header;

a memory including a plurality of memory regions; and

a processor operative to receive the frame with the dummy portion from the alignment module and store said frame in the memory, the dummy portion ~~shifting~~ aligning the header and payload portion in the memory such that the first bit of the payload is aligned on a boundary between adjacent memory regions.

2. (Original) The apparatus of claim 1, wherein the alignment module is operative to prefix the dummy portion to the header.

3. (Original) The apparatus of claim 1, wherein the alignment module is operative to suffix the dummy portion to the header.

4. (Original) The apparatus of claim 1, wherein the network interface comprises a switch.

5. (Original) The apparatus of claim 1, wherein the frame comprises an Ethernet frame.

6. (Original) The apparatus of claim 1, wherein the payload comprises an Internet Protocol (IP) packet.

7. (Original) The apparatus of claim 1, wherein the header includes n bits and the memory regions include m bits, wherein the ratio n/m has a non-zero remainder p.

8. (Original) The apparatus of claim 7, wherein the dummy portion includes p bits.

9. (Original) The apparatus of claim 7, wherein n is 112.

10. (Original) The apparatus of claim 7, wherein m is 32.

11. (Original) The apparatus of claim 7, wherein p is 16.

12. (Original) The apparatus of claim 7, wherein the dummy portion consists of x bits, where $x = m*c + p$ and c is an integer.

13. (Original) The apparatus of claim 1, wherein the processor is operative to access the frame header in a shifted position in the memory.

14. (Original) The apparatus of claim 1, further comprising a protocol stack to extract the payload from the frame in the memory.

15. (Original) The apparatus of claim 14, wherein the protocol stack comprises a TCP/IP software stack.

16. (Currently Amended) A method comprising:

receiving a frame including a header and a payload, the payload having a first bit adjacent the header;

appending a dummy portion to the header; and

storing the frame with the dummy portion in a memory with a plurality of memory regions, the dummy portion ~~shifting~~ aligning the header and payload in the memory such that the first bit of the payload is aligned on a boundary between adjacent memory regions.

17. (Original) The method of claim 16, wherein said appending comprises prefixing the dummy portion to the header.

18. (Original) The method of claim 16, wherein said appending comprises suffixing the dummy portion to the header.

19. (Original) The method of claim 16, wherein said receiving the frame comprises receiving the frame at a switch.

20. (Original) The method of claim 16, wherein the frame comprises an Ethernet frame.

21. (Original) The method of claim 16, wherein the payload comprises an Internet Protocol (IP) packet.

22. (Original) The method of claim 16, wherein the header includes n bits and the memory regions include m bits, wherein the ratio n/m has a non-zero remainder p.

23. (Original) The method of claim 22, wherein the dummy portion includes p bits.

24. (Original) The method of claim 22, wherein n is 112.

25. (Original) The method of claim 22, wherein m is 32.

26. (Original) The method of claim 22, wherein p is 16.

27. (Original) The method of claim 22, wherein the dummy portion consists of x bits, where $x = m*c + p$ and c is an integer.

28. (Original) The method of claim 16, further comprising accessing the frame header in a shifted position in the memory.

29. (Original) The method of claim 16, further comprising extracting the payload with a protocol stack.

30. (Original) The method of claim 29, wherein the protocol stack comprises a TCP/IP software stack.

31. (Currently Amended) An apparatus comprising:

means for receiving frames, each frame including a header and a payload, the payload having a first bit adjacent the header;

means for appending a dummy portion to the header;

a memory including a plurality of memory regions; and

means for receiving the frame with the dummy portion ~~from the alignment module~~ and storing said frame in the memory, the dummy portion ~~shifting~~ aligning the header and payload portion in the memory such that the first bit of the payload is aligned on a boundary between adjacent memory regions.

32. (Original) The apparatus of claim 31, the means for appending comprises means for prefixing the dummy portion to the header.

33. (Original) The apparatus of claim 31, the means for appending comprises means for suffixing the dummy portion to the header.

34. (Original) The apparatus of claim 31, wherein the network interface comprises a switch.

35. (Original) The apparatus of claim 31, wherein the frame comprises an Ethernet frame.

36. (Original) The apparatus of claim 31, wherein the payload comprises an Internet Protocol (IP) packet.

37. (Original) The apparatus of claim 31, wherein the header includes n bits and the memory regions include m bits, wherein the ratio n/m has a non-zero remainder p.

38. (Original) The apparatus of claim 37, wherein the dummy portion includes p bits.

39. (Original) The apparatus of claim 37, wherein n is 112.

40. (Original) The apparatus of claim 37, wherein m is 32.

41. (Original) The apparatus of claim 37, wherein p is 16.

42. (Original) The apparatus of claim 37, wherein the dummy portion consists of x bits, where $x = m*c + p$ and c is an integer.

43. (Original) The apparatus of claim 31, further comprising means for accessing the frame header in a shifted position in the memory.

44. (Original) The apparatus of claim 31, further comprising means for extracting the payload from the frame in the memory.

45. (Original) The apparatus of claim 44, wherein the means for extracting comprises a TCP/IP software stack.